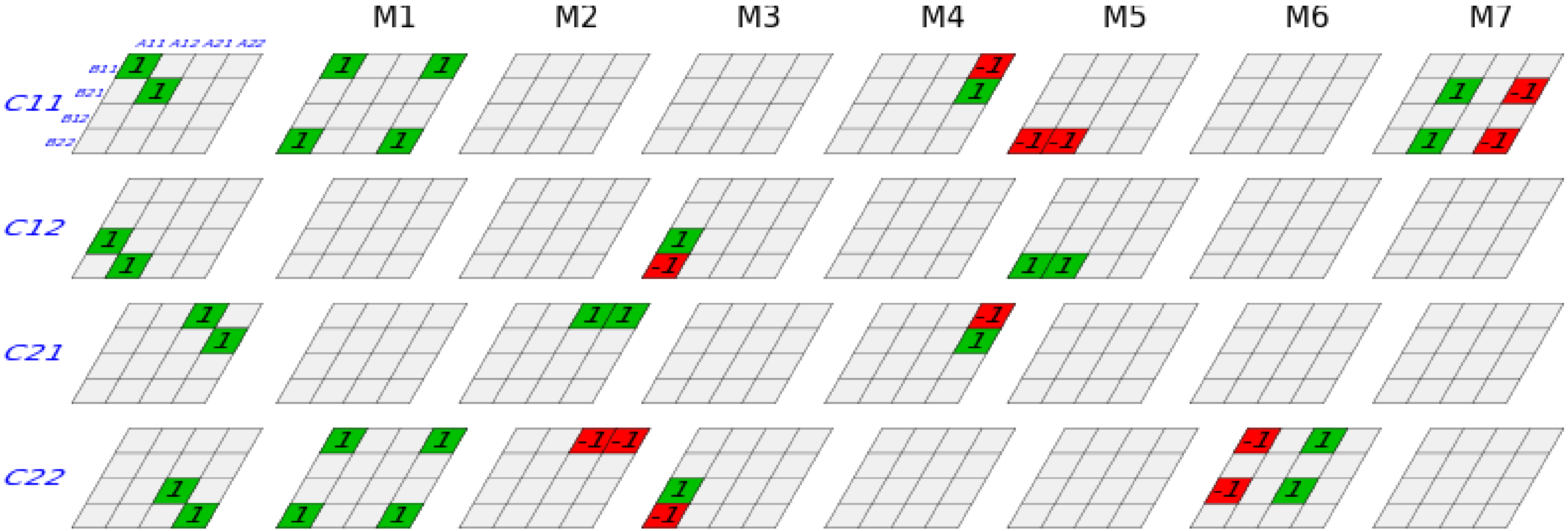
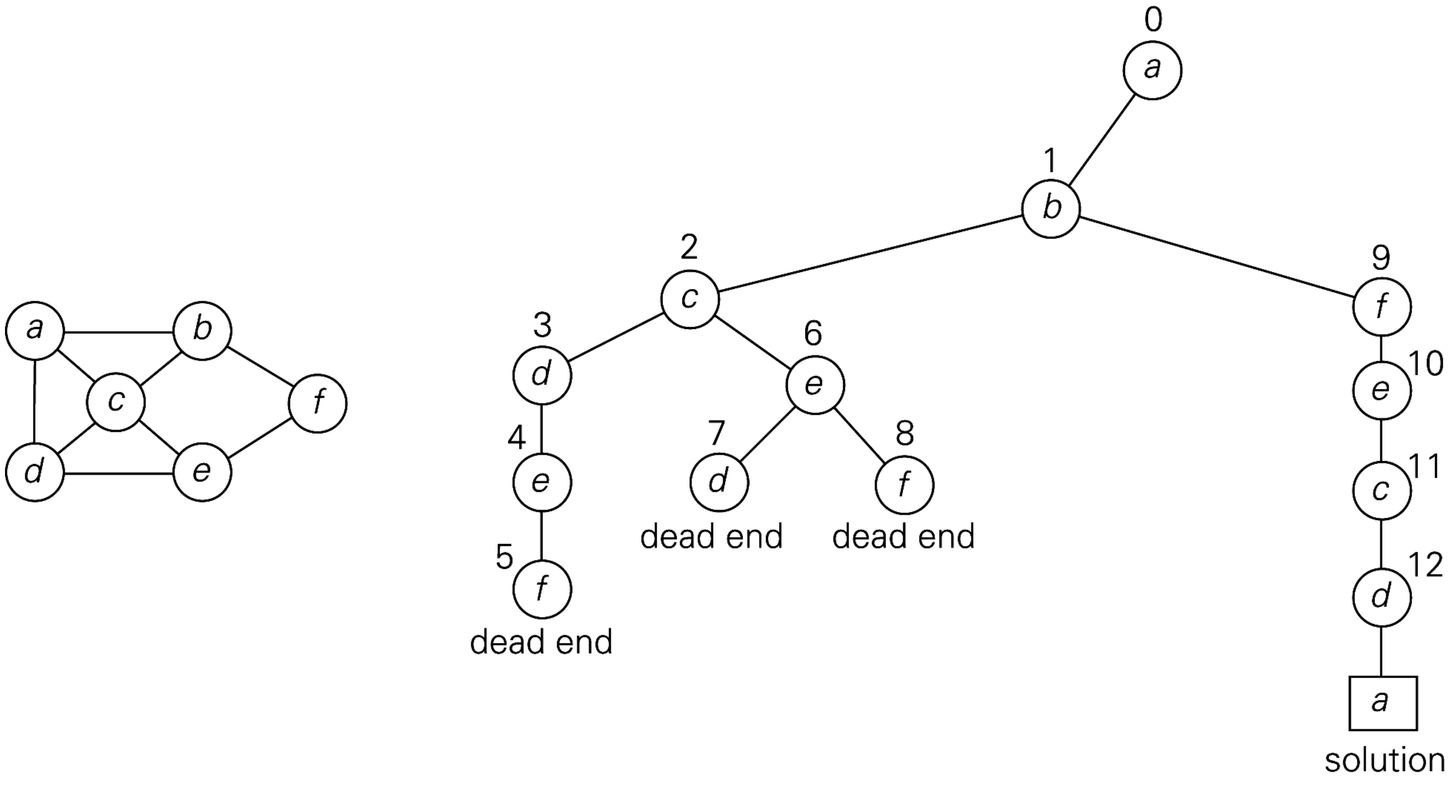
Backtracking Sum of Subset Problem

V. Balasubramanian

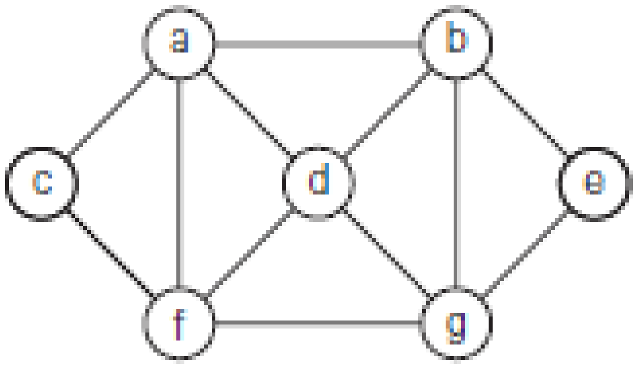


M1
M2
M3
M4
M5
M6
M7
A11 A12 A21 A22
B11
B21
B12
B22
C11
C12
C21
C22

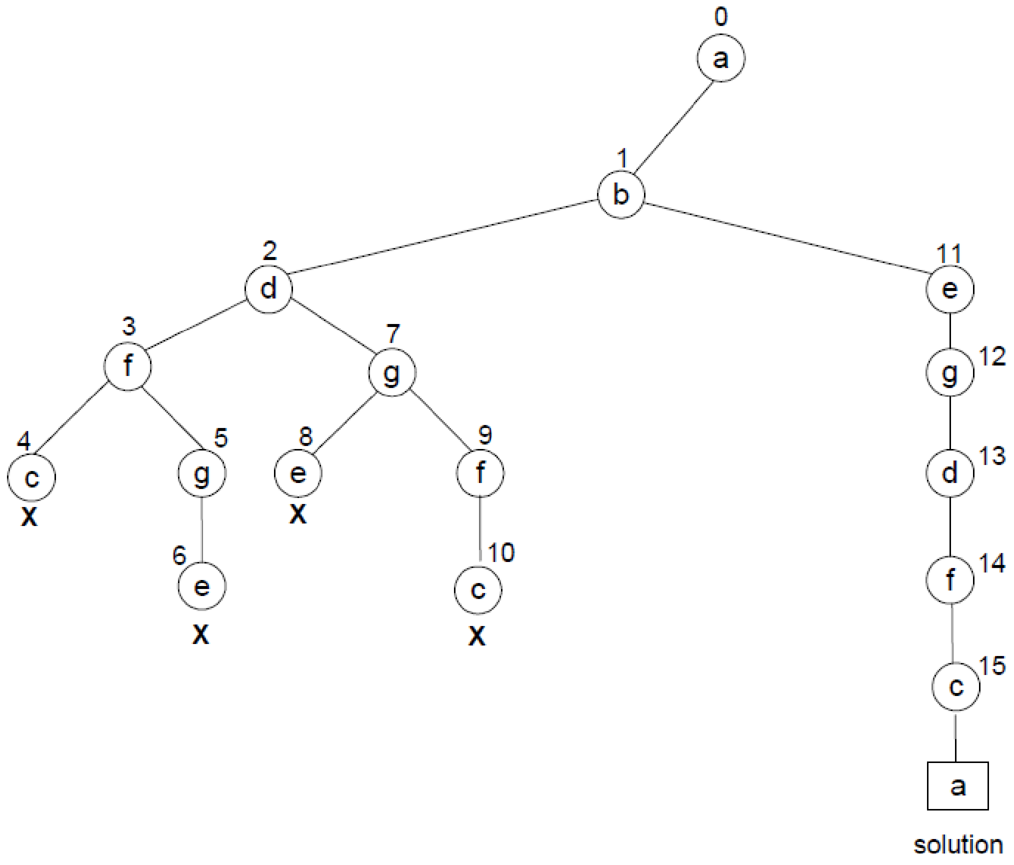


(a) (b)

FIGURE 12.3 (a) Graph. (b) State-space tree for finding a Hamiltonian circuit. The numbers above the nodes of the tree indicate the order in which the nodes are generated.


a
b
c
d
e
f
g

Solution



0
a
1
b
2
d
11
e
3
f
7
g
12
g
4
c
X
5
g
8
e
X
9
f
13
d
6
e
X
10
c
X
14
f
15
c
a
solution

Sum of Subset problem

Recall our thief and the 0-1 Knapsack Problem from Section 4.4.1. In this problem, there is a set of items the thief can steal, and each item has its own weight and profit. The thief's knapsack will break if the total weight of the items in it exceeds W. Therefore, the goal is to maximize the total value of the stolen items while not making the total weight exceed W. Suppose here that the items all have the same profit per unit weight. Then an optimal solution for the thief would simply be a set of items that maximized the total weight, subject to the constraint that its total weight did not exceed W. The thief might first try to determine whether there was a set whose total weight equaled W, because this would be best. The problem of determining such sets is called the Sum-of-Subsets Problem.

Example

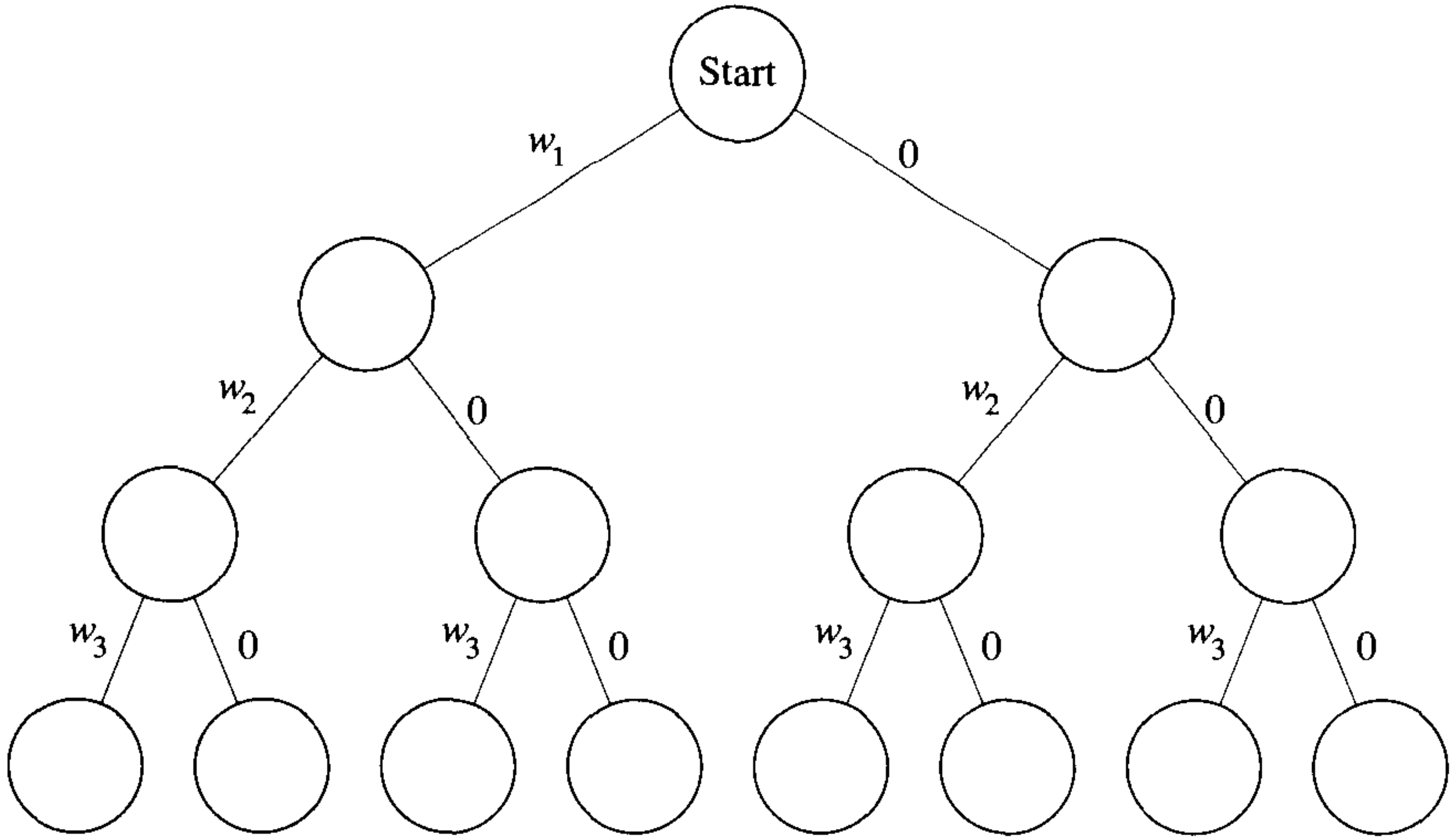
Suppose that $n = 5$, $W = 21$, and

$$w_1 = 5 \qquad w_2 = 6 \qquad w_3 = 10 \qquad w_4 = 11 \qquad w_5 = 16.$$

Because

$$w_1 + w_2 + w_3 = 5 + 6 + 10 = 21$$
$$w_1 + w_5 = 5 + 16 = 21$$
$$w_3 + w_4 = 10 + 11 = 21,$$

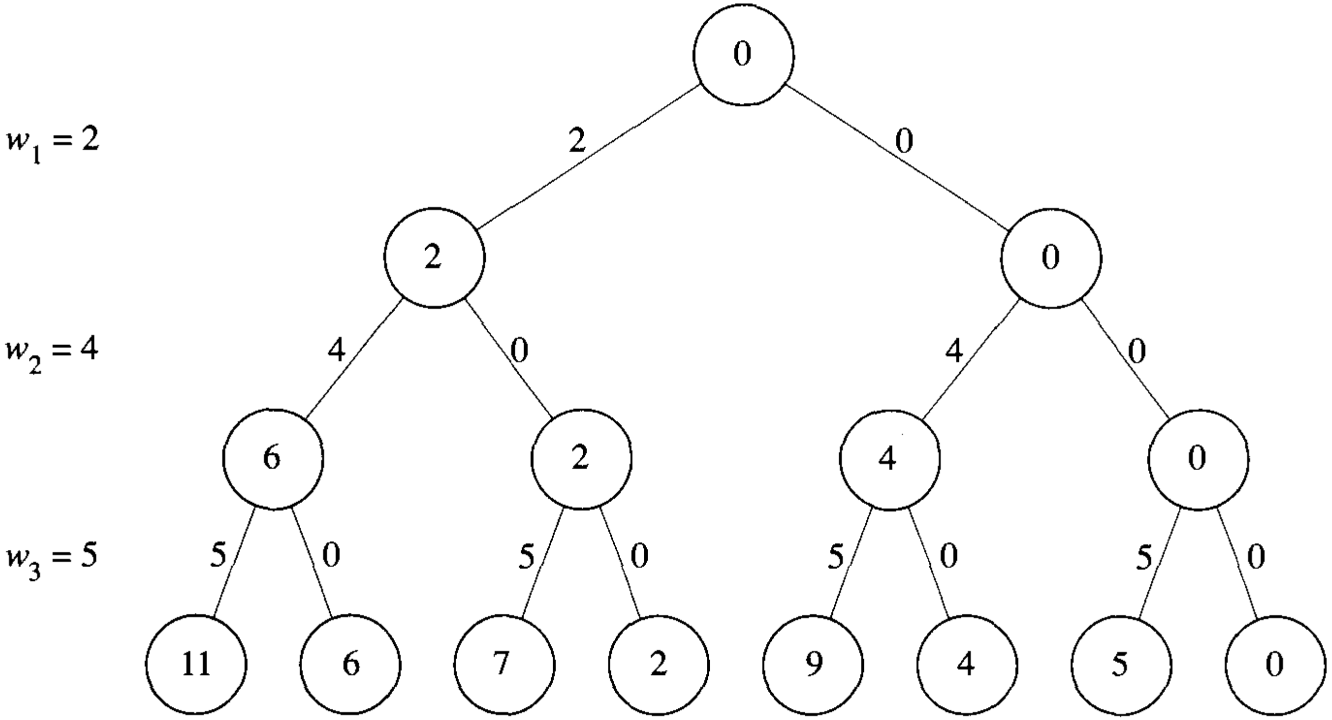
the solutions are $\{w_1, w_2, w_3\}$, $\{w_1, w_5\}$, and $\{w_3, w_4\}$.



Start
w_1
0
w_2
0
w_2
0
w_3
0
w_3
0
w_3
0
w_3
0

Figure 5.8 shows the state space tree for $n = 3$, $W = 6$, and

$$w_1 = 2 \qquad w_2 = 4 \qquad w_3 = 5.$$



It is convenient to sort the set's elements in increasing order. So, we will assume that

$$a_1 < a_2 < \cdots < a_n.$$

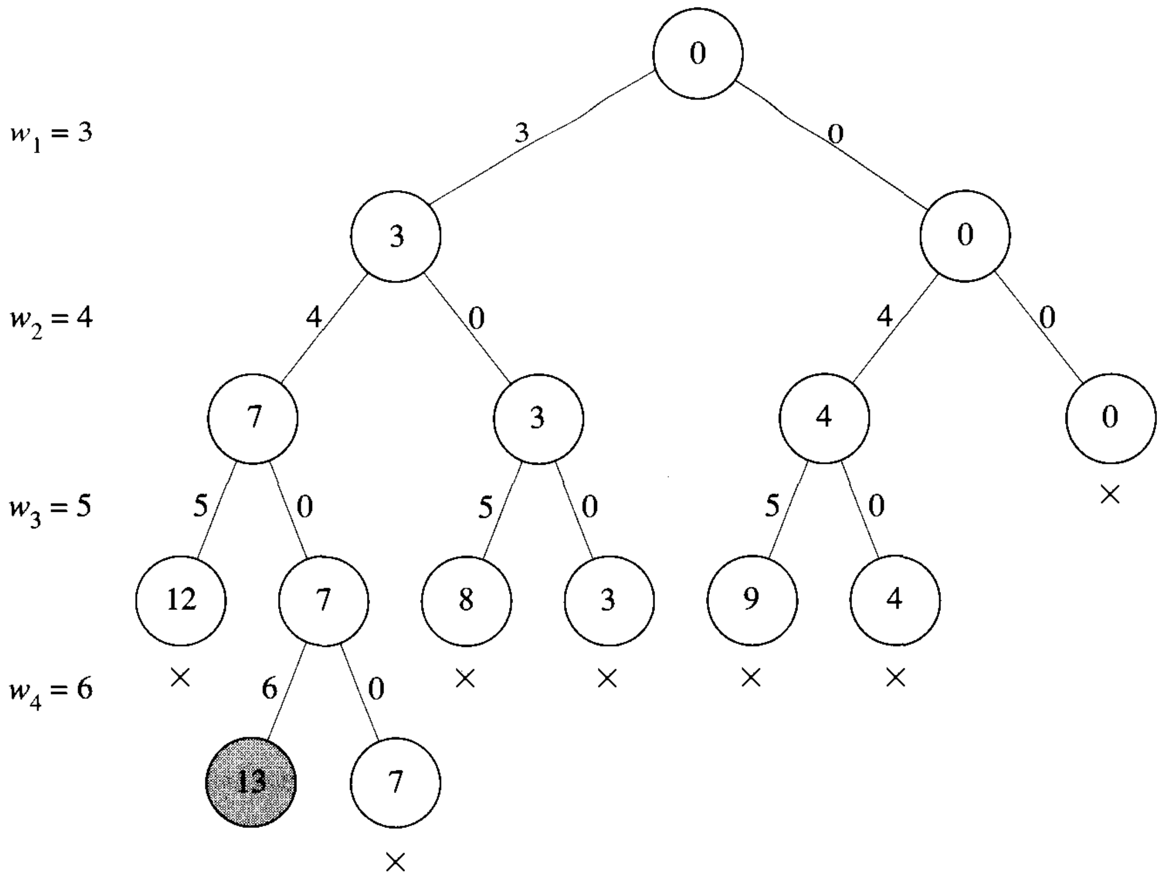
$$weight + w_{i+1} > W.$$

$$weight + total < W.$$

Example

$n = 4$, $W = 13$, and

$$w_1 = 3 \qquad w_2 = 4 \qquad w_3 = 5 \qquad w_4 = 6.$$



$w_1 = 3$
$w_2 = 4$
$w_3 = 5$
$w_4 = 6$
0
3
0
3
0
4
0
4
0
7
3
4
0
5
0
5
0
5
0
12
7
8
3
9
4
6
0
13
7

$$s + a_{i+1} > d \quad \text{(the sum } s \text{ is too large)},$$

$$s + \sum_{j=i+1}^{n} a_j < d \quad \text{(the sum } s \text{ is too small)}.$$

Example 2

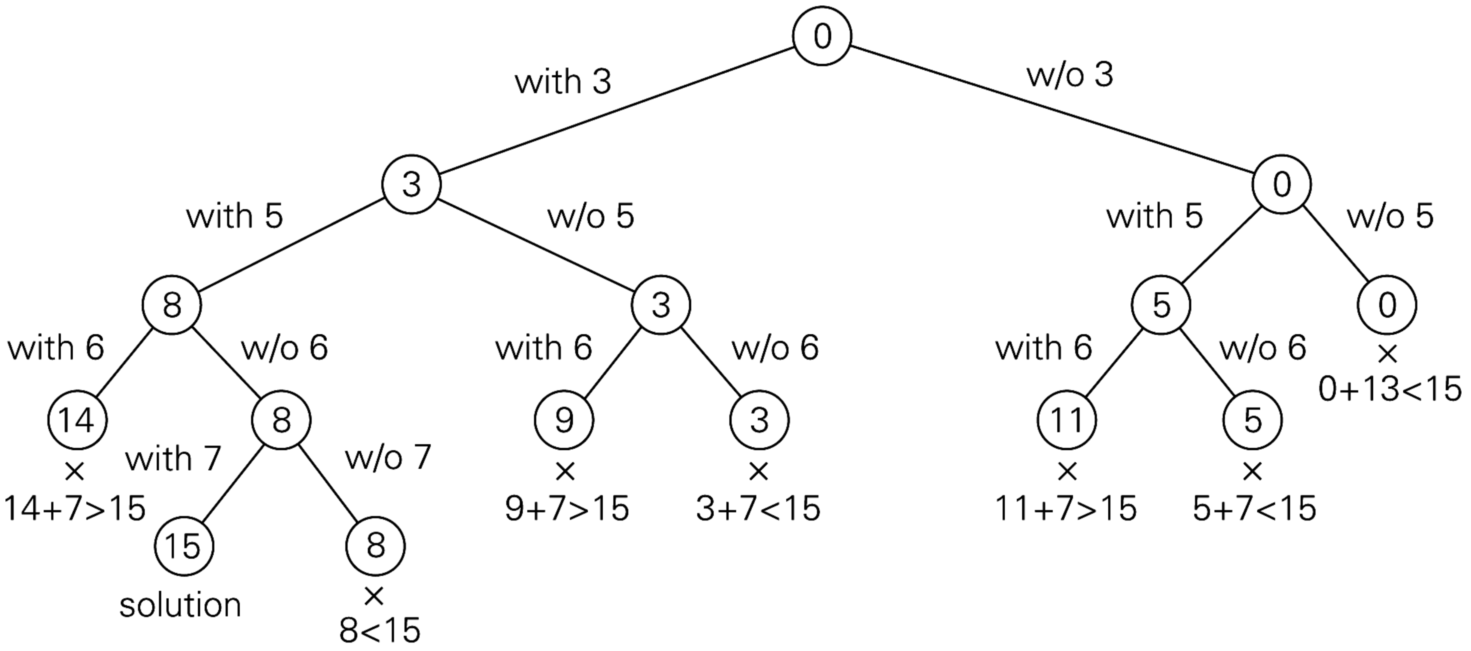
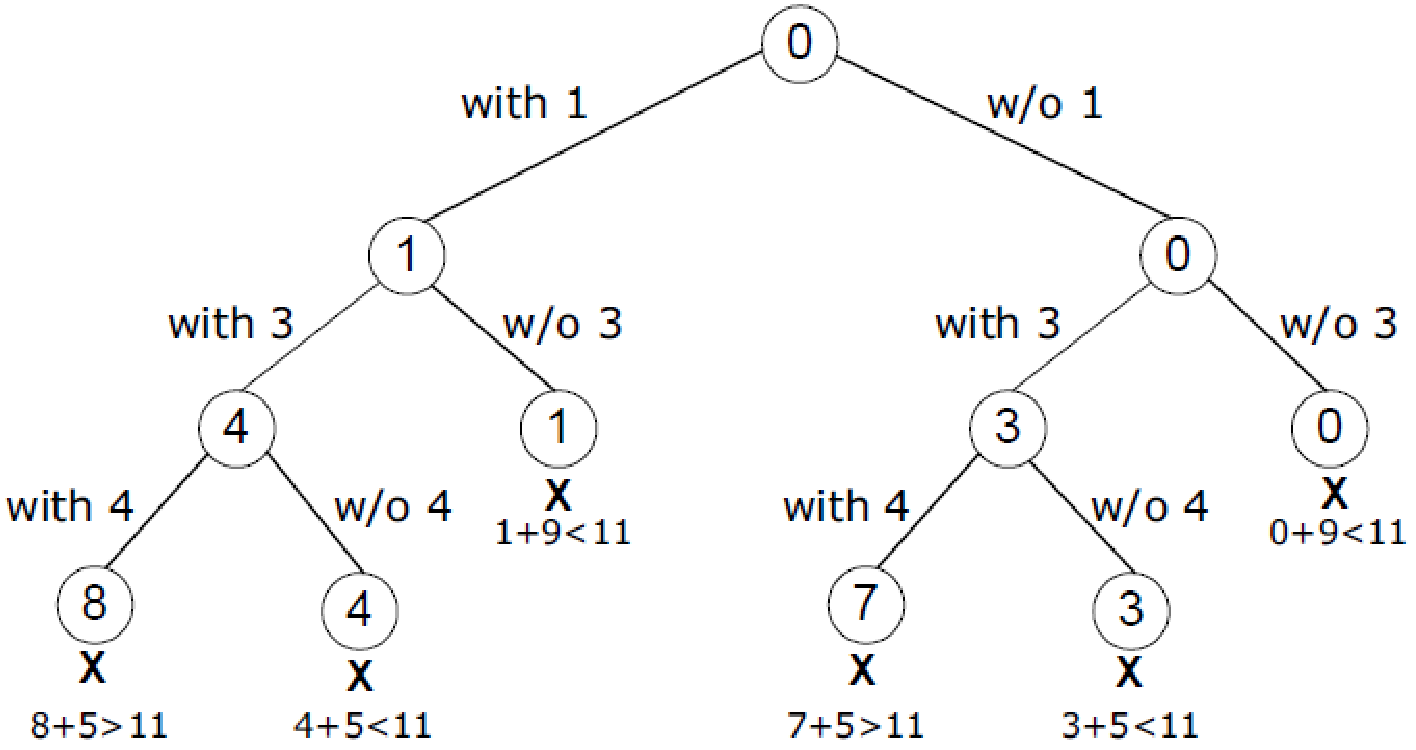


FIGURE 12.4 Complete state-space tree of the backtracking algorithm applied to the instance $S = \{3, 5, 6, 7\}$ and $d = 15$ of the subset-sum problem. The number inside a node is the sum of the elements already included in subsets represented by the node. The inequality below a leaf indicates the reason for its termination.

a. Apply backtracking to solve the following instance of the subset sum problem: $A = \{1, 3, 4, 5\}$ and $d = 11$.

b. Will the backtracking algorithm work correctly if we use just one of the two inequalities to terminate a node as nonpromising?


0
with 1
w/o 1
1
0
with 3
w/o 3
with 3
w/o 3
4
1
3
0
with 4
w/o 4
X
1+9<11
with 4
w/o 4
X
0+9<11
8
4
7
3
X
X
X
X
8+5>11
4+5<11
7+5>11
3+5<11